# Вывод полей/подполей

Вывод поля/подполя начинается либо с префикс-литерала (условного или повторяющегося, можно пустого), либо с непосредственного обращения к полю.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Префикс-литералы** | | | | **Обращение к полю** | **Суффикс-литералы** | |
| "Условный" | Команды горизонтального и вертикального размещения | Команды смены режима | |Повторяющийся| | V200^A | |Повторяющийся| | "Условный" |
| Произвольное количество, могут перемешиваться друг с другом (однако начинаться всё должно условным литералом) | | | Не более одного | Не более одного | |
| Все команды между первым условным префикс-литералом и соответствующей ему командой вывода поля/подполя становятся условными и будут выполнены только при наличии поля/подполя, иначе они игнорируются. | | | Если такой литерал есть, то он должен непосредственно предшествовать соответствующей ему команде вывода поля/подполя. | Если такой литерал есть, то он должен следовать непосредственно за соответствующей ему командой вывода поля/подполя. | Если такой литерал есть, то он должен следовать непосредственно за повторяющимся суффикс-литералом или связанной с ним командой вывода поля/подполя. |
| Суффикс-литералы не должны разделяться запятыми и не должно быть запятой между командами вывода поля/подполя и первым суффикс-литералом, так как запятая обозначает конец суффикс-литералов, связанных с данной командой вывода поля/подполя. | |
| Литералы подвергаются преобразованию в прописные буквы, если расположены после соответствующей команды режима вывода. | | | | | | |

Если после условного или повторяющегося префикс-литерала не следует обращения к полю, он становится «сиротским», т. е. не выполняется никогда.

# Обращение к полю

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Команда** | **Тег** | **Индекс** | **Вложенное** | **Подполе** | **Индекс** | **Смещение** | **Длина** | **Параграф** | |
| **Первая строка** | **Другие строки** |
| Обязательные | | Необязательные | | | | | | | |
| **V** | Целое число | [индекс] | @целое число | ^символ | [индекс] |  |  | (целое число | , целое число) |
| **G** |  |  |
| **D** | Неприменимо | | | |
| **N** |

Заметим, что можно использовать специальный разделитель подполей ^\* для вывода первого подполя, если оно есть. В этом случае, первому подполю необязательно должен предшествовать его разделитель.

# Индекс

Нумерация повторений полей/подполей начинается с единицы.

Спецификация индекса – один из вариантов:

1. Целое число-литерал, например, 2 – второе повторение.
2. Арифметическое выражение (можно использовать переменные и функции), например, $x + 2.
3. \* - последнее повторение.
4. + - новое повторение (имеет смысл только в операторе присвоения значения полю/подполю).
5. Выражение1 .. выражение2 – все повторения от «выражение1» до «выражение2» (включая его). Оба выражения могут быть арифметическими.

Если индекс не указан, то:

1. Для поля:
   1. Внутри повторяющейся группы считается равным текущему номеру повторения группы;
   2. Вне группы – все повторения поля.
2. Для вложенного поля – первое повторение.
3. Для подполя – первое повторение.

# Присвоение полю/подполю

Предусмотрен оператор присвоения значения полю/подполю.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Элемент оператора** | Спецификация поля/подполя | Знак равенства | Формат | Точка с запятой (ограничитель) |
| **Пример** | V300[2] | = | 'Язык текста: ', v101 | ; |

Отрицательные значения индекса отсчитываются от последнего повторения, т. е. -1 означает последнее повторение, -2 – предпоследнее и т. д.

Нулевое значение индекса генерирует ошибку времени исполнения и приводит к остановке скрипта.

Примеры оператора присвоения:

|  |  |
| --- | --- |
| **Оператор** | **Выполняемое действие** |
| V200='^aЗаборостроение^eучебник'; | Все имеющиеся повторения удаляются, создаётся столько повторений, сколько строк выдаёт формат |
| V300[2]='Язык текста: ', v101; | На место указанного повторения помещается результат, выданный форматом. Если формат выдаёт более одной строки, последующие повторения сдвигаются вниз. |
| V300[2..3]=('Язык текста: ', v101[1..2], /); | На место указанных повторений помещается результат, выданный форматом. Если строк недостаточно, последующие повторения сдвигаются вверх. Если строк оказывается больше, последующие повторения сдвигаются вниз. Пустые строки не создают новых повторений поля/подполя. |
| V300[\*]='Последнее повторение поля'; | Установка значения последнего повторения поля. |
| V300[nocc(v300)-1]='Предпоследнее повторение поля'; | Индекс вычисляется арифметическим выражением. |
| V300[+]='Новое повторение поля'; | Добавление нового повторения поля. |
| V200^a[2]='Внезапно!'; | Установка (если потребуется – создание) второго повторения подполя a в поле 200. Выполняется для всех повторений поля 200. |
| V200[1]^a[2]='Внезапно!' | Установка второго повторения подполя a в первом повторении поля 200. |
| V300=V300[1]/V300[3..]; | По факту, удаляется второе повторение поля 300 (если оно было). |

Присвоение пустой строки (пустого формата) означает удаление соответствующего повторения поля/подполя, например:

V300[2]=;

Если происходит обращение к несуществующему повторению поля/подполя, то создаётся новое повторение (при непустом присваиваемом значении), при этом актуальное значение индекса может оказаться отличающимся от указанного.

Пример скрипта, который «переворачивает» повторения поля 300:

$temp = (v300/);  
$len = size ($temp);  
for $i = 1; $i <= $len; $i = $i + 1;  
do  
 v300[$i] = $temp[$len - $i + 1];  
end

Можно обойтись без временного массива:

$len = size ((v300/)); $len2 = $len / 2;  
for $i = 1; $i <= $len2; $i = $i + 1;  
do  
 v300[$len - $i + 1] = $v300[$i];  
end

# Цикл for

'Почему это круто?'#

'================='##

for $x=0; $x < 10; $x = $x+1;

do

$x, ') ',

'Прикольно же!'

#

end

##

'По выходу из цикла x=', $x

Вложенные циклы:

for $x=0; $x < 10; $x = $x+1;

do

$x, ' '

for $y=0; $y < $x; $y = $y+1;

do

'='

end

#

end

# Цикл ForEach

foreach $x in v200^a,v200^e,'Hello'

do

$x

#

end

Вложенные циклы

foreach $x in v200^a, v200^e

do

$x, ' => '

foreach $y in (v702^a/)

do

$y, ' ',

end

#

end

# Цикл while

'Почему это круто?'#

'================='##

$x=0;

while $x < 10 do

$x, ') ',

'Прикольно же!'

#

$x=$x+1;

end

##

'По выходу из цикла x=', $x

Вложенные циклы

$x=0;

while $x < 10

do

$x, ' ',

$y=0;

while $y < $x

do

'=',

$y=$y+1;

end

#

$x=$x+1;

end

# Стандартные функции

Bold

Cat

Chr

CommandLine

COut

Debug

Error

Fatal

GetEnv

Include

IOcc

Italic

Len – длина строки в символах (не байтах!)

MachineName

NOcc

Now

NPost

OSVersion

Size – длина массива (фактически – количество строк в результате отработки формата)

Sort – сортировка массива

System – выполнение команды операционной системы

Today –

ToLower

ToUpper

Trace

Trim

Warn